International Journal of Computer and Electrical Engineering, Vol. 2, No. 1, February, 2010
1793-8163

Digital Combinational Circuits Design By QCA Gates

Pijush Kanti Bhattacharjee, *Member, IACSIT*

***Abstract*—Different logic gates like MV, NOT, AOI, NNI etc under QCA nanotechnology are introduced. NNI gate is highly effective regarding space and speed consideration. Any Boolean functions are synthesized by MV and NNI gates or simply NNI gates alone, eliminating inverter (NOT) gate. A new method for realizing adder circuit in binary reversible logic is invented. This procedure synthesizes for a more efficient designing adder circuits with little "garbage". The method is used in a classical logic. I use only 2-inputs and 2-outputs AND-NAND and OR-NOR cells designed on QCA technology. It admits a recursive construction. Ultimately a general equation for the minimum number of gates required to an arbitrary number of input variables, causing synthesis of adder circuits, is achieved. It provides a significant reduction in hardware cost and switching delay compared to the other existing techniques. All other type of combinational circuits like subtractor, multiplier, divisor, multiplexer, encoder, comparator etc can be designed by the adder circuit only.**

***Index Terms*—Adder Circuit Design, 2-Inputs and 2-Outputs AND-NAND (A-NA) gate, And-Or-Inverter (AOI) gate, Majority Voter (MV) gate, Moore's Law, Nand-Nor-Inverter (NNI) gate, 2-Inputs and 2-Outputs OR-NOR (O-NO) gate**.

I. Introduction

Reversible Quantum Computers (QCs) will be created for improvement in speed, reducing cost and space etc [1]-[7]. Reversible are the circuits or the gates that have the same number of inputs and outputs and have one-to-one mappings between vectors of inputs and outputs; thus the vector of the input states can be uniquely reconstructed from the vector of the output states.

Fredkin Gate (FG) is a fundamental concept in reversible and quantum computing, the base of "realization-related" papers. It has been introduced by Ed Fredkin and Tomasso Toffoli in 1982 [5] and explained as mentioned below:

(i) Every Boolean function can be build from (binary) Fredkin Gates (FGs), such that it has three inputs A, B and C and three outputs P, Q and R like

P = A

Q = if A then C else B

R = if A then B else C

(ii) Feyman Gates is "Controlled NOT" or "quantum XOR", such gates have two inputs A and B and two outputs P and Q, they are called linear likewise

P = A

Q = A EXOR B

Thus synthesis of adder Boolean circuits either half adder or full adder circuits are achieved by using a simple recursive arrangement of 2-inputs, 2-outputs AND-NAND (A-NA) and OR-NOR (O-NO) logic gates realizable with QCA (Quantum Dot Cellular Automata) technology. It eliminates the use of NOT or inverter gate which is more costly and spacious as well as introduced delay in realizing the adder circuit. By this approach, adder circuits for an arbitrary number of input variables are achieved. A general equation for estimating the number of gates (AND-NAND, OR-NOR etc) required with garbage outputs for adder circuits realization for any arbitrary number of input variables is invented. Further, the proposed technique is applicable for any adder circuits as well as combinational circuits synthesis results a logic design with less hardware cost which drastically reduces the "garbage" compared to the other existing techniques.

II. Quantum Dot Cellular Automata

QCA (Quantum-dot Cellular Automata) [1]-[12] devices encode and process binary information as charged arrays of charge coupled quantum dots. A quantum cell can be viewed as a set of four charge containers or dots positioned at the corners of a square, as shown in Fig. 1. It contains two extra mobile electrons. The electrons can quantum mechanically tunnel between dots but can not come out from the cell and are forced to settle at the corner positions due to coulomb interaction. Thus, there exists two equivalent energetically minimal arrangements for the electrons in a QCA cell (Fig. 1), i.e. the polarization P = +1 (representing logic 1) and P = –1 (representing logic 0).

In Fig. 1, a QCA cell and its binary logic are shown, the energetically position of the diagonal electrons identifies the binary logic 0 or 1. This phenomenon is useful in nano technology which affects high resolution fast electronic circuits.

Pijush Kanti Bhattacharjee is an Assistant Professor in the Department of Electronics and Communication Engineering, Bengal Institute of Technology and Management, Santiniketan, P.O. Doranda, West Bengal, Pin-731236, India. He was an Ex Asssitant Director in the Department of Telecommunications (DoT), Government of India, India. He has possessed vast working experience in the field of Telecommunications including Mobile Communications, Image Processing, VLSI etc during last 29 years.
He is a member of IACSIT, Singapore; CSTA, USA; IAEng, Hongkong. (phone: +91-33-25954148; email: pijushbhatta_6@hotmail.com)..

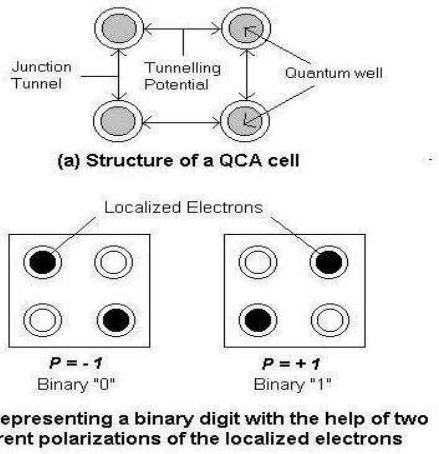


Fig.1 A QCA cell and its binary logic

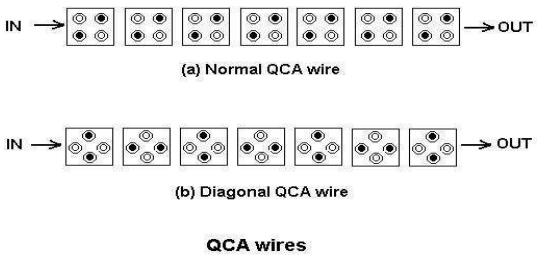


Fig.2 Information propagating through QCA wires.

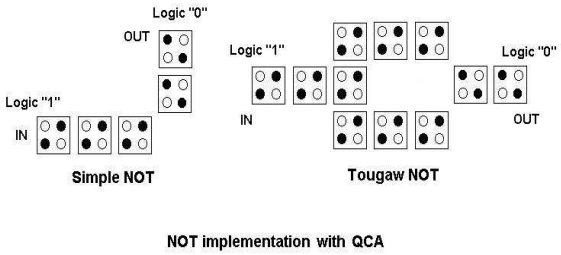


Fig.3 Implementation of NOT with QCA Gate.

In this power consumption for changing the charge of electron is very much less compare to that of general charge carriers (hole-electron) electronic components. A QCA Cell with its binary logic creates a new direction in nano technology [1]-[12]. It requires minimum current or energy to change any state i.e. previous state. Thus, a minimum recurring cost for power consumption is effective in this QCA gates which is highly applicable in advanced super fast processors. At the same time the data manipulation rate i.e. data processing rate is very fast comparing to the conventional processors designed by CMOS gates in VLSI technology. Also power or heat dissipation, electro magnetic wave radiation etc are very much less in QCA based gates using nano technology.

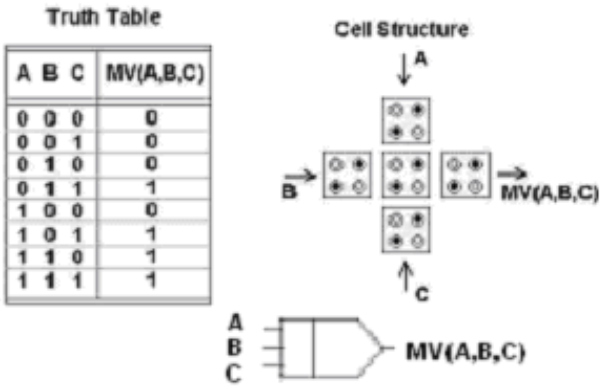


Fig.4 QCA Majority Voter (MV) Gate

The basic QCA logic elements [1]-[10] include a QCA wires are shown in Fig. 2, QCA inverter or NOT gate in Fig. 3 and Majority Voter (MV) or Majority Gate (Maj) in Fig. 4. In diagonal (also called 45^0) QCA wire, which is shown in Fig. 2(b), binary signal alternates polarization in successive cells. The QCA Majority Voter (MV) shown in Fig. 4 realizes MV(A, B, C) = Maj(A, B, C) = AB+BC+CA, outputs '1' if there are two or more 1s in an input pattern. The classical AND and OR gates can be realized with the majority gate by fixing an input as 0 and 1 respectively. The majority gate is not a universal gate. It can not realize the logical NOT operation. The functionally complete set is {MV, NOT}. Thus all Boolean functions can not be synthesized by MV gates alone, but MV with NOT gates is used to ascertain the completeness.

Therefore, the designers have to use separate QCA cell arrangement for realization of the logical NOT. The 5-input (A, B, C, D and E) And-Or-Inverter (AOI) gate [8] with embedded AND, OR and INV functions has been proposed to provide the universal gate function. However the AOI suffers from the limitation of proper separation of input and output binary wires- that is, in fixing the distances d1, d2 and d3 of Fig. 5.

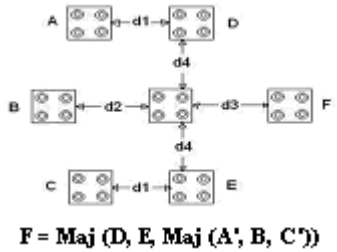


Fig.5 QCA And-Or-Inverter (AOI) Gate.

Another drawback of A-O-I gate is that it requires more space and the A-O-I gates are more complex nature comparing to that of MV or NOT gates.

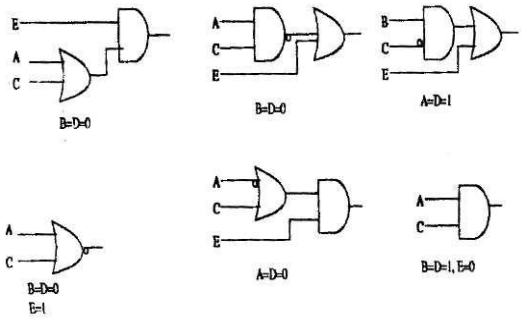
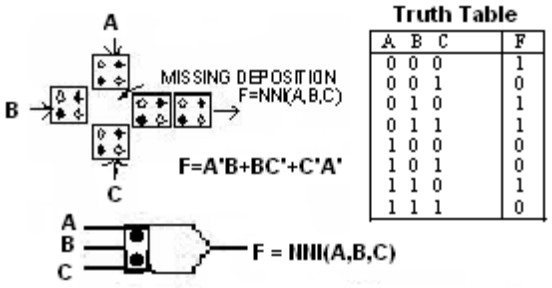


Fig.6 Various logic functions realized with AOI Gate



Truth Table

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Fig.7 QCA Nand-Nor-Inverter (NNI) Gate

Thus to implement MV with NOT functions, a new gate called Nand-Nor-Inverter (NNI) [12] is constructed, where NNI(A, B, C) = MV(A', B, C') = A'B+BC'+C'A'. NNI gate is a majority voter (MV) gate having two inverted inputs. NNI gate construction is clearly shown in Fig. 7. The NNI gate is a universal gate and can be employed for realizing versatile logic functions. It proves to be as effective as the AOI (And-Or-Inverter) gate and requires lesser overhead, for setting the variables, than that of an AOI, while realizing the basic logic gates. NNI gate ensures very less space comparing to that of the other gates like MV, AOI and inverter (NOT) gate. It is explained vigorously that all Boolean functions which can be constituted within thirteen number of universal Boolean functions [8], are realized either by combination of MV and NNI gates or by NNI gates alone [12].

III. Combinational Circuits Design

A Combinational circuit consists of input variables, logic gates and output variables. The logic gates accept signals from the inputs and generate the outputs which are different logic function combinations of the inputs.

Firstly we design the adder circuit, then all other Boolean functional circuits like subtractor, multiplier, divisor, comparator, multiplexer, demultiplexer, encoder, decoder, differentiator, integrator etc can be easily constructed by the adder circuit alone.

A. Adder Circuit Design

The adder circuits design both half adder and full adder circuit synthesis is paying attention to the researchers in the field of VLSI and nanotechnology design, especially for logic synthesis. A number of synthesis techniques for Boolean adder circuits design are reported [3], [11]. I represent a simple cost effective adder circuits design with QCA gates for future generation digital design.

B. Half Adder Circuit Design

Half adder circuit is designed conventionally by simple digital logic gates i.e. by Exclusive-Or (EX-OR), AND gates etc. When two binary inputs A and B are added, according to the half adder truth table, the sum and the carry are written as mentioned below:

Sum, S1 = A (+) B = AB'+A'B, where (+) is exclusive or (XOR) function or gate, Carry, T1 = AB. This half adder circuit is also designed by CMOS gates as well as QCA gates like MV, NOT, AOI, NNI, AND-NAND, OR-NOR gates etc.

C. Full Adder Circuit Design

Full adder circuit is implemented by digital logic gates. If three binary inputs A, B and C are adding, the full adder outputs i.e. the sum and the carry at par the truth table are as:
Sum, S2 = A (+) B (+) C = AB'C'+A'BC'+A'B'C+ABC,
Carry, T2 = AB + BC + CA.
Generally one full adder circuit is constructed by the two half adder circuits as shown in Fig. 8.

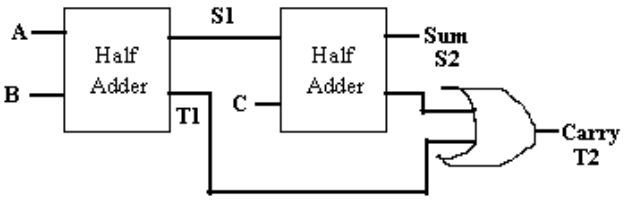


Fig.8 Full Adder Circuit Designed by Two Half Adder Circuits

Adder circuits designing by CMOS gates are described by Baker et. al. [3]. For half adder implementation with two binary inputs, number of CMOS transistors requiring for sum (S1) is 6 numbers (including NOT gate i.e. two CMOS gates for inverting inputs) and for carry (T1) is 2 numbers. Thus total CMOS gates necessary for the half adder circuit are 8. In case of Full Adder design using three binary inputs by CMOS transistors framing AOI types [3], number of CMOS gates requiring for sum (S2) are 8 and for carry (T2) are 6. Therefore total numbers of CMOS gates require for the full adder circuit with three binary inputs (numbers) are 14. If the same full adder circuit is constructed using half adder circuits by recursive implementation as shown in Fig. 8, the total number of CMOS gates require 19 (8 CMOS gates for each half adder circuit and 3 CMOS gates for OR function). Accordingly for addition of four number binary bits (inputs), the total CMOS gates require 30 numbers.

Full adder circuits designed with MV and NOT gates by Walus and Jullien [11] are shown in Fig. 9. It consists of three number MV gates and two number NOT (Inverter) gates. Thus this full adder circuit with three binary inputs is comprising with total five gates which are all based on QCA technology. It severely reduces the requirement of total gates and circuitry as described in earlier cases. For making half adder circuit from this full adder circuit, one input is taken as zero.

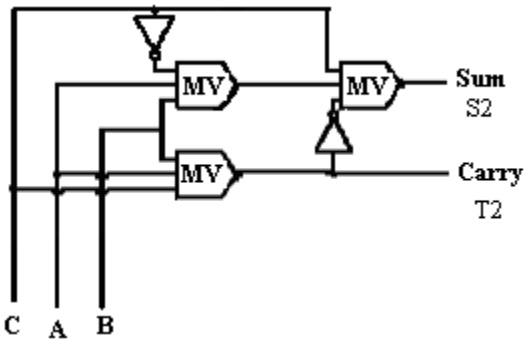


Fig.9 Full Adder Circuit Diagram by MV and NOT gates

A new QCA gates called NNI gate is invented [12], we have modified this full adder circuit by using MV and NNI gates, which is clearly explained in Fig. 10. We mathematically evaluate the implementation of full adder circuit design by using MV and NNI gates from existing MV and NOT gates,

Sum, S2 = MV(MV(A, B, C'), MV(A', B', C'), C)
or, S2 = MV(MV(A', B', C'), C, NNI(A', C', B'))
or, S2 = NNI(MV(A, B, C), C, NNI(A, C, B))
Carry, T2 = AB + BC + CA = MV(A, B, C)

Therefore, this full adder circuit comprising with MV and NNI gates has one MV gate and two NNI gates.

The area of an MV, NOT, AOI and NNI in 20 nm X 20 nm cell technology (with quantum dot size of 5 nm) are 75 nm X 75 nm, 125 nm X 75 nm, 125 nm X 115 nm and 100 nm X 75 nm respectively. We consider 5 nm separation between two neighboring QCA cells (cell to cell distance) for MV, NOT, NNI and for the AOI as shown in Fig. 5, $d_1 = d_3 = 25$ nm and $d_2 = 35$ nm [8].

For comparison, the area of a realization is computed in terms of the area of a majority gate (A_{mv} i.e. 75 nm X 75 nm). Thus the total area taken by the full adder circuit designed by MV and NOT gates as shown in Fig. 9 is 6.333 and that of MV and NNI gates as shown in Fig. 10 is 3.666.

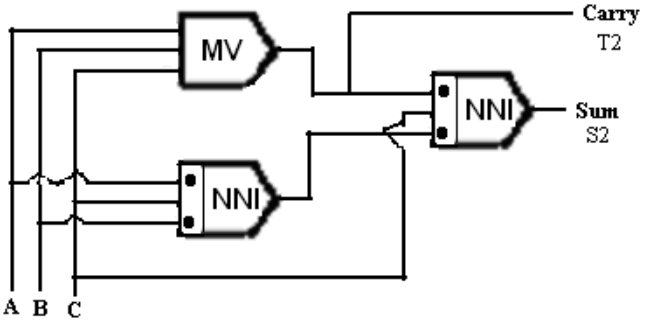


Fig.10 Full Adder Circuit Diagram by MV and NNI gates

Therefore the modified full adder circuit with MV and NNI gates is more efficient regarding space and speed consideration. I also formulate a general equation for full adder circuit synthesis for any arbitrary binary numbers (inputs) adding regarding the number of gates required as well as the garbage outputs.

D. Subtractor Circuit Design

Subtraction of binary numbers is performed by taking the complement of the subtrahend and adding to the minued as the same manner in addition. It is also possible to subtract binary numbers directly using logic gates. If the minued bit is smaller than the subtrahend bit, a '1' is borrowed from the next higher significant bit position and it is treated as the input for the next higher stage.

Half subtractor logic according to the truth table is like,
Subtract, D1 = X (+) Y = XY'+X'Y and Borrow, B1 = X'Y, where X is minued and Y is subtrahend bit.
Full subtractor logic is also from the truth table like,
Subtract, D2 = X (+) Y (+) Z = XY'Z'+X'YZ'+X'Y'Z+XYZ,
Borrow, B2 = YZ + X'Y + X'Z.

The difference between adder and subtractor circuits is that Carry (T) and Borrow (B) differs by inverting the first input. Hence the same circuit with certain inputs can be used adder as well as subtractor circuits by using a control input.

IV. Methodology And Implementation

The proposed method of adder circuits design and its implementation are discussed in this section. I propose new QCA gates e.g. AND-NAND (A-NA) and OR-NOR (O-NO) gates [12]. The output of an AND-NAND gate is AND and NAND functions of the inputs. Similarly, the output of an OR-NOR gate is OR and NOR functions of the inputs. The gate is having two inputs and two outputs. These two gates are the simplest gates of QCA cells under nano technology. I eliminate the use of NOT or inverter gate which takes more area, costlier, time and delay consuming.

In the synthesis of adder circuits, result of adding number of input variables is obtained in binary (digital) quantity. This becomes a generalized solution for the adder circuit realization for any number of input variables.

A. AND-NAND & OR-NOR QCA Gates

First of all, I take 2-inputs and 2-outputs AND-NAND, OR-NOR gates as shown in Fig. 11. Here, AND-NAND gate and OR-NOR gate are two type of QCA gates which are designed on nano technology process.

These AND-NAND, OR-NOR gates based on QCA technology require less overhead area and at the same time ensure best efficiency in speed.

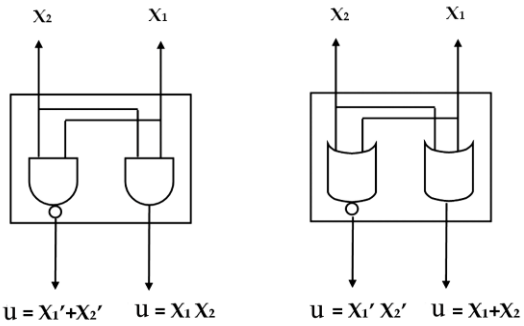


Fig.11 2-inputs and 2-outputs AND-NAND Gate (left), 2-inputs and 2-outputs OR-NOR Gate Circuits (right).

B. Half Adder Circuit (2-Inputs) Design

For 2-binary input variables adding, we take one AND(A)-NAND(NA) gate and two OR(O)-NOR(NO) gates to get the Sum (S1) as well as the Carry (T1).

The realization of half adder circuit by AND(A)-NAND(NA) and OR(O)-NOR(NO) gates is shown in Fig. 12. These QCA based gates are taking minimum

space area as well as minimum power consumed to formulate any type of adder circuit outputs. Therefore, these adder circuit realization integrated circuits may be manufactured at a minimum cost surprising all other type of invented ICs, especially when it is manufactured at large scale. This type of half adder circuit design is employed in all other complex circuit realization also e.g. full adder circuit synthesis etc.

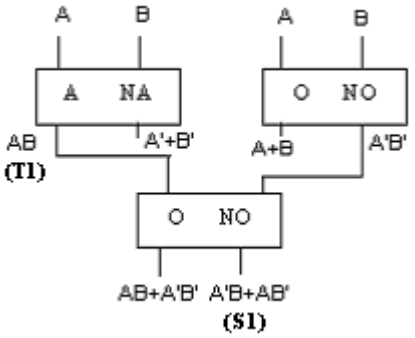


Fig.12 Half Adder Circuit Diagram by AND-NAND & OR-NOR.

For two input variables, total three gates are required in which one AND-NAND gate and two OR-NOR gates. Three outputs from all these gates are the garbage outputs which is not belonging to the half adder outputs i.e. either the sum or the carry. Therefore, this construction of half adder circuit by these QCA gates is the simplest form.

C. Full Adder Circuit (≥ 3-Inputs) Design

Full adder circuit by employing two half adder circuits is constructed as described in Fig. 8. In the proposed technique, I use AND-NAND & OR-NOR QCA gates only [12]. Three binary numbers (digits) for designing full adder circuit are added. It requires total seven QCA gates, out of which two number AND-NAND gates and five number OR-NOR gates are used as shown in Fig. 13. In this case, the garbage outputs are seven in number, marked as G1 to G7 in Fig. 13. This full adder circuit is very less cost effective and space minimized. Now for addition of an extra binary number to this three digit full adder i.e. four binary number addition full adder circuit, additional four number QCA gates are required to this 3-digit full adder circuit, in which, there consists of one AND-NAND gate and three OR-NOR gates. Therefore 4-digits full adder circuit design, total eleven number of QCA gates are required, this composing of three number AND-NAND gates and eight number OR-NOR gates. Similarly, the garbage outputs for 4-binary digits full adder circuit are 11 numbers. The same 4-binary digits full adder circuit is constructed by either 30 number CMOS gates or 11 number MV, NOT gates or 7 number MV, NNI gates as shown in Table-I. Likewise, I design full adder circuit for any arbitrary number of binary (digital) inputs, accordingly the sum and the carry are obtained as output. Therefore, I formulate a general equation for addition of any number of binary (digital) inputs i.e. full adder circuit design regarding

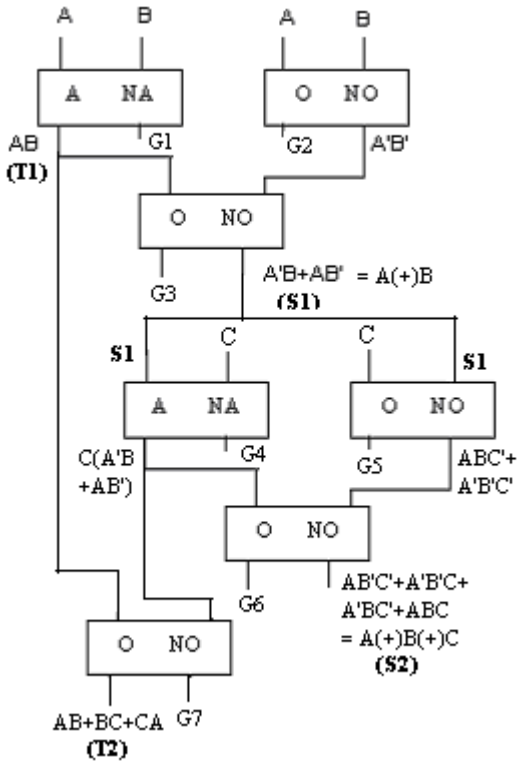


Fig.13 Full Adder Circuit Diagram by AND-NAND & OR-NOR QCA gates with Three Binary Inputs.

the number of gates required and the garbage outputs as clearly described in Table-I.

Likewise multiplexer, demultiplexer, encoder (Decimal to Octal, Binary to Hexadecimal etc), decoder, code converter (BCD to Gray, Gray to ASCII etc) and different logic combination functions can be well realized by the adder circuit only.

V. CONCLUSIONS

Although adder circuits design, half adder circuit as well as full adder circuits are difficult to realize practically with the minimum number of gates, but my proposed design with adoption of QCA technology enlightens a new direction in the field of adder circuit synthesis. This type of adder circuits manufactured with QCA gates specially AND-NAND, OR-NOR gates require very much less number of gates. Hence it increases enormous switching speed as well as reducing cost and space. Therefore these proposed QCA gates AND-NAND, OR-NOR gates are able to design adder circuits with any arbitrary number of binary inputs. Ultimately a general solution for minimum number of QCA gates (AND-NAND and OR-NOR) requiring is invented, identifying the "garbage" output. In coming years, Moore's Law is only satisfied by adopting my invented adder as well as combinational circuits designed with special QCA gates like A-NA, O-NO gates. Thus QCA based design of adder circuit are simulated by AND-NAND, OR-NOR gates and compared with the results of conventional AND, OR, EX-OR, CMOS gates and other QCA gates like MV, NOT, NNI etc in Table-I. It is also observed that the implementation of adder circuit design with QCA based AND-NAND and OR-NOR gates is more advantages than that of general logic gates

design like AND, OR, NOT, EX-OR, CMOS etc and QCA gates like MV, NOT, NNI etc. In all aspects, this QCA based design adder circuits are an excellent tool in higher growth of IC processors having different combinational circuits. Thus it can afford as a precious element in the forthcoming generation.

REFERENCES

[1] C. S. Lent, P. D. Taugaw, W. Porod and G. H. Berstein, "Quantum Cellular Automata", *Nanotechnology*, vol. 4, no. 1, January 1993, pp 49-57.

[2] Z. Kohavi, *Switching and Finite Automata Theory*, Tata McGraw Hill Pub Company Ltd, New Delhi, 2nd Edition, 2007.

[3] R. J. Baker, H. W. Li, D. E. Boyce, *CMOS Circuit Design, Layout and Simulation*, Prentice-Hall of India Pvt. Ltd, Eastern Economy Edition, New Delhi, 2004.

[4] I. Amlani, A . O. Orlov, G. Toth, C. S. Lent, G. H. Bernstein and G. L. Sinder, "Digital Logic Gate using Quantum Dot Cellular Automata", *Science*, vol. 284, no. 5412, April 1999, pp. 289-291.

[5] E. Fredkin, T. Toffoli, "Conservative Logic", *International Journal of Theoretical Physics*, vol. 21, no. 3-4, April 1982, pp 219-253.

[6] A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent and G. L. Sinder,"Realization of a Functional Cell for Quantum Dot Cellular Automata", *Science*, vol. 277, no. 5328, August 1997, pp 928-930.

[7] C. S. Lent, P. D. Taugaw, "A Device Architecture for Computing with Quantum Dots", *Proceedings IEEE*, vol. 85, no. 4, April 1997, pp. 541-557

[8] M. Momenzadeh, M. B. Tahoori, J. Huang and F. Lombardi, "Characterization, Test and Logic Synthesis of AND-OR-INVERTER (AOI) Gate Design for QCA Implementation", *IEEE Trans on Computer Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, December, 2005, pp. 1881-1893.

[9] R. Zhang, K. Walus, W. Wang and G. A. Jullien, "A Method of Majority Logic Reduction for Quantum Cellular Automata", *IEEE Trans on Nanotechnology*, vol. 3, no. 4, Dec 2004, pp. 443-450.

[10] M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. S. Lent, G. H. Bernstein, G. L. Snider and F. Peiris, "Quantum Dot Cellular Automata at a Molecular Scale", *Annals of the New York Academy of Sciences*, vol. 960, 2002, pp. 225-239.

[11] K. Walus, G. A. Jullien, "Design Tools for an Emerging SoC Technology: Quantum Dot Cellular Automata", *Proceedings of the IEEE*, vol. 94, no. 6, June 2006, pp. 1225-1244.

[12] P. K. Bhattacharjee, "Efficient Synthesis of Symmetric Functions", *World Congress in Computer Science, Computer Engineering and Applied Computing (WORLDCOMP-2008) in International Conference on Computer Design (CDES'08),* las Vegas, USA, July 2008, pp. 23-29.

Dr. Pijush Kanti Bhattacharjee is a pioneer in Engineering, Management, Law, Indo-Allopathy, Herbal, Homeopathic and Yogic medicines. He is having qualifications M.E, MBA, MDCTech, A.M.I.E, B.Sc, B.A, LLB, BIASM, CMS, PET, EDT, FWT, DATHRY, B.Mus, KOVID, DH, ACE, FDCI etc. He worked Department of Telecommunications (DoT), Govt. of India from June 1981 to Jan 2007 (26 years), lastly holding Assistant Director post at RTEC [ER], DoT, Kolkata, India. Thereafter, he worked at IMPS College of Engineering and Technology, Malda, WB, India as an Assistant Professor in Electronics and Communication Engineering Department from Jan,2007 to Feb,2008 and Feb, 2008 to Dec, 2008 at Haldia Institute of Technology, Haldia, WB, India. In Dec, 2008 he joined at Bengal Institute of Technology and Management, Santiniketan, WB, India in the same post and department. He has written two books "Telecommunications India" & "Computer". He is a member of IACSIT, Singapore; CSTA, USA; IAEng, Hongkong and IE, India.

Table – I
Comparison of Adder Circuit Designed by Any Arbitrary Number of Input Variables with Different CMOS and QCA Gates

Srl No	Number of Binary Input Variables for Designing Adder Cicuit	Total Number of CMOS gates	Total Number of MV, NOT gates	Total Number of MV, NNI gates	Total Number of AND-NAND Gates	Total Number of OR-NOR Gates	Total No of AND-NAND, OR-NOR gates	Total No of Garbage Output not act as further Input for AND-NAND, OR-NOR Gates
i)	2 (Half Adder)	8	5 (3 MV, 2 NOT)	3 (1 MV, 2 NNI)	1	2	3	3
ii)	3 (Full Adder)	19 (Half Adder) , 14 (AOI)	5 (3 MV, 2 NOT)	3 (1 MV, 2 NNI)	2	5	7	7
iii)	4	30	11 (7 MV, 4 NOT)	7 (3 MV, 4 NNI)	3	8	11	11
iv)	5	41	17 (11 MV, 6 NOT)	11 (5 MV, 6 NNI)	4	11	15	15
v)	*n*	*8 + 11(n−2)*	*For n=2,5 For n >3 (all others), 5 + 6(n−3)*	*For n=2, 3 For n >3 (all others), 3 + 4(n−3)*	*n−1*	*2(n−1) + (n−2)*	*3 + 4(n − 2)*	*3 + 4(n − 2)*